Generative Multisimulation: Decision-Support under Uncertainty using Evolutionary Multimodels

Levent Yilmaz Bradley Mitchell

Auburn University

Cargese Interdisciplinary Seminar, Cargese, France

Outline

Cyber Physical Systems

- Cyber-physical systems (CPS) - tight conjoining of and coordination between computational and physical resources.
- CPS requirements: adaptability, autonomy, efficiency, reliability, safety.
- CPS characteristics
    - Systems that respond more quickly (e.g., autonomous collision avoidance),
    - more precise (e.g., robotic surgery and nano-tolerance manufacturing),
    - work in dangerous or inaccessible environments (e.g., autonomous systems for search and rescue, firefighting, and exploration),

CPS capabilities

CPS needs capabilities that facilitate deeply embedding

- computational intelligence,
- communication, control, and coordination

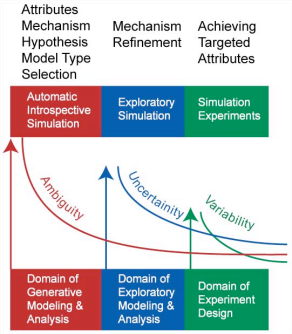
for

- sensing, actuation, and adaptation

into physical systems with

- active and reconfigurable components.

Coping with Uncertainty and Ambiguity



- Focus in prediction, optimization, (performance and representational) efficiency, scalability etc.
- Partial consideration of uncertainty (lack of information) and
- Little discussion on ambiguity (lack of clarity)

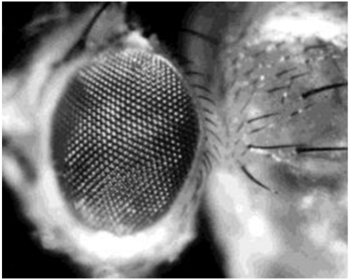
Decision-support under Uncertainty

- Need methods to improve robustness and resilience of decisions.
- Experimenting with evolutionary and/or contingency models in real-time in unstructured problems with the characteristics of
  - deep uncertainty,
  - dynamic environments, and
  - shifting, illdefined, and competing goals.
- Need ability to observe the system in real-time and adapt useful characteristics with as little computational effort as possible

The Basis for Autonomic Generative Multisimulation

- Need to cope with an unpredictable environment autonomously through **flexibility** and robustness.
- Flexibility can be achieved using different but closely related approaches:
  - **Adaptation**. The system changes its behavior through learning and adaptation.
  - **Anticipation**. Next state depends on its current state as well as the current image(s) of its future state(s).

The Compound Arthropod Eye Analogy



Compound Eye Metaphor

- Excellent at detecting motion. As an object moves across the visual field, ommatidia are progressively turned on and off.
- Because of the "flicker effect", insects respond far better to moving objects (e.g., situations) than stationary ones.

Major Goals and Questions

- What are different forms of uncertainty and
- How can they be modeled in such a way to facilitate exploration and variation of configurations in an efficient manner?
- Can the key principles of CAS evolution be leveraged to attain robustness.

Solution Approach

Rather than rely on a single authoritative model, GMS explores an ensemble of plausible models:

- individually flawed but collectively provide more insight
- insights derived from the model ensemble are used to improve the performance of the system under study
- as the system develops, observations improve exploration of the model ensemble.
- In essence, a useful co-evolution between the physical system and GMS occurs

Principles

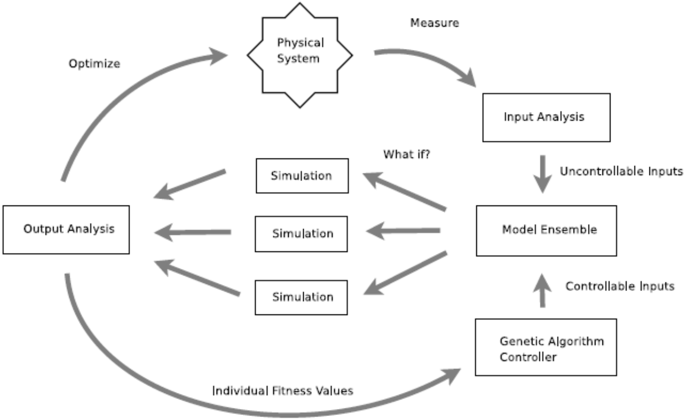
Model schema (agents) undergo three types of change:

- first order change - adapt the observation to the existing schema; second order change - purposeful change in the schema in order to better fit observations; and third order change - schema survives or dies because of the survival or death of its corresponding fitness.

Schema change makes model ensembles more

- robust (it can perform in light of increasing variation or variety),
- reliable (it can perform more predictably), or
- grow in requisite variety (in can adapt to a wider range of conditions).

Hybrid Exploration



To efficiently search a potentially infinite number of plausible models, AMS uses a hybrid exploration technique.

Hybrid Exploration

- Uncontrollable inputs and controllable inputs are handled with input analysis and output analysis modules.
- System observations are used to hypothesize distributions for uncontrollable inputs.
- As details emerge, fidelity of distributions improve.
- Controllable input factors representing the configuration of the physical system are evolved using a genetic algorithm.

Uncontrollable Factors				Controllable Factors			
X_{11}	X_{12}	$\cdots$	X_{1m}	G_1	G_2	$\cdots$	G_ℓ
X_{21}	X_{22}	$\cdots$	X_{2m}	G_1	G_2	$\cdots$	G_ℓ
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
X_{n1}	X_{n2}	$\cdots$	X_{nm}	G_1	G_2	$\cdots$	G_ℓ

Table: A Combined Model Ensemble of n models is composed of a Partial Model Ensemble of m sampled variables, X, and an individual of ℓ genes, G. Each row represents a single model for which one or more simulation replications should be performed. Note that the same individual is used in each model.

Local Adaptation

- In GMS, Particle Swarm Optimization (PSO) is used for implementing the learning element of adaptive agents.
- PSO as a continuous numeric optimization technique in which a potential solution to a problem is characterized as a point in some n-dimensional space, with the number of dimensions being equal to the number of decision variables

Particle Swarm Optimizer

$$\overrightarrow{v}_{id} = \chi(v_{id} + c_1\epsilon_1(p_{id} - x_{id}) + c_2\epsilon_2(p_{gd} - x_{id})) \tag{1}$$

Global Adaptation

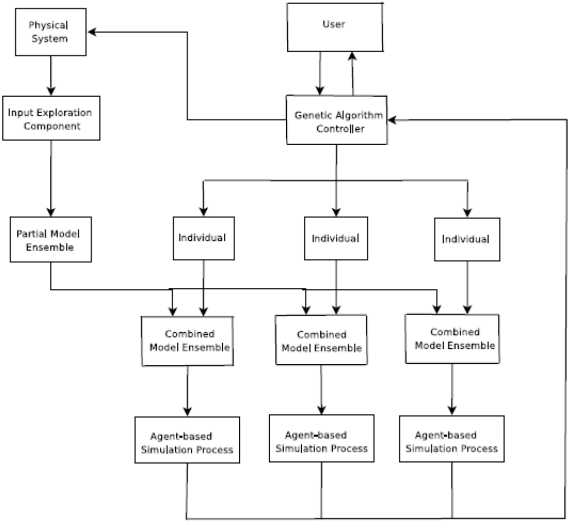
- AMS evolves potential system configurations for use within the physical system.
- Search of the space of potential system configurations requires the exploitation of the performance characteristics of the configuration.

Evolutionary Computation

Algorithm 1 : A genetic algorithm.

```
pop[m] ⇐ createInitialPopulation( )
for i = 1 to m do
    pop[i] ⇐ calculateFitness( pop[i] )
end for
repeat
    children[n]
    for i = 1 to n do
        parents[2] ⇐ selectParents( pop )
        children[i] ⇐ crossover( parents )
        children[i] ⇐ mutatation( children[i] )
        children[i] ⇐ calculateFitness( children[i] )
    end for
    pop[m] ⇐ selectSurvivors( pop, children )
until termination = true
```

GMS Component Architecture



GMS Component Architecture

- Input Exploration Component (IEC) - input analysis and selecting appropriate distributions for the uncontrollable model factors.
- Samples are used to create a PME, a copy of which is integrated with each individual to form one CME.
- The Genetic Algorithm Controller (GAC) is responsible for evolving the population of individuals - used to form the CMEs. and a number of Agent Simulation Instances (ASPs) to simulate CMEs in parallel.
- Ideally, each ASP is mapped to one or more CPU cores.

Component Interaction Dynamics

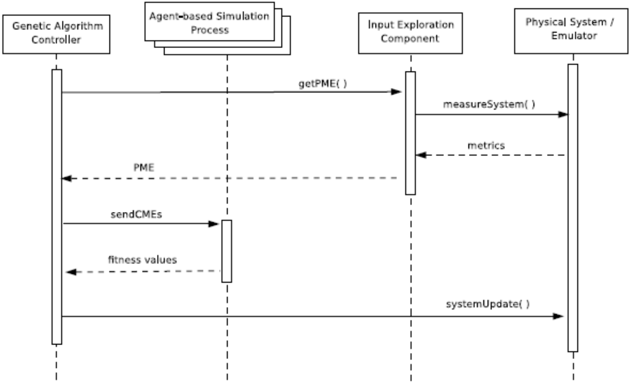


Figure: Component Interaction in GMS

UAV Search, Rescue, Attack: Scenario

- Based on an agent-based model of an autonomous UAV team in a Search and Attack mission.
- Interact through local communication only. There is no global system of coordination or prior intelligence of targets.
- A rule-based approach for UAV movement is implemented. e.g., steering behaviors
- Takes place in a 2-dimensional space of equal dimensions.
    - UAVs start from a base located at the center of the map.
    - Distributed randomly across the map and their positions are initially unknown to the UAVs. Once they are launched, the UAVs must find and destroy all of the targets as quickly as possible.

Initial State

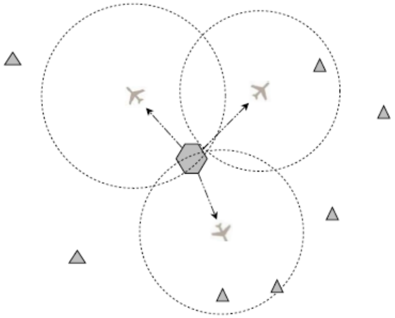
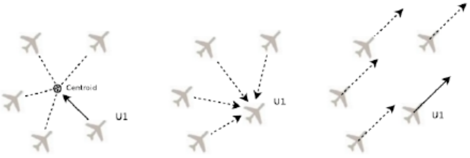


Figure: The Search and Attack scenario shortly after simulation start. UAVs depart from the centrally located airbase (represented by a hexagon) with uniformly random initial movement vectors. The sensor envelopes of individual UAVs are represented by dotted circles.

UAV Movements



(a) Cohesion (b) Separation (c) Alignment

Figure: The cohesion, separation, and alignment rules are useful for coordinating group behaviors. Cohesion causes a UAV to move toward the average position, or centroid. Separation causes a UAVs to move away from other UAVs. alignment causes a UAV to align its velocity vector with the velocities of other UAVs

UAV Decision Rules

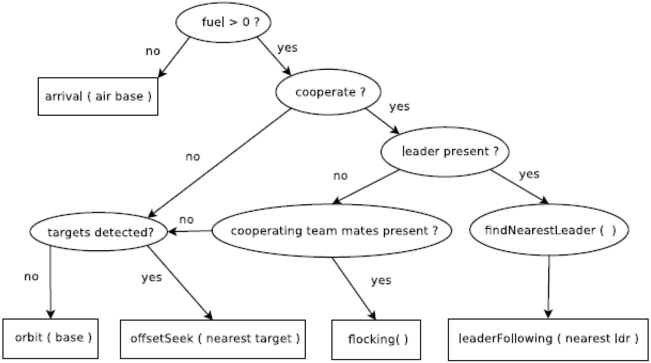


Figure: Decision Making Progress.

Optimization

$$max(x) = 1 - \frac{1}{a^{(k+1)} + 2} - \frac{1}{d + 2} \tag{2}$$

- The fitness of a particle's position varies from 0 to 1 and is determined by maximizing the objective function.
  - *a* is the number of attacks performed by the UAV,
  - *k* is the number of the UAV's kills, and
  - *d* is the number of target detections propagated by the UAV
- the UAV prefers those locations in the optimization space that favor
  - number of attacks and UAV kills,
  - with increased capability to help others through large number of potential message exchanges.

Implementation

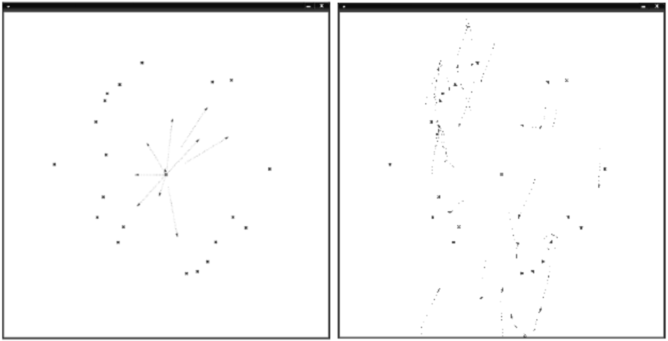


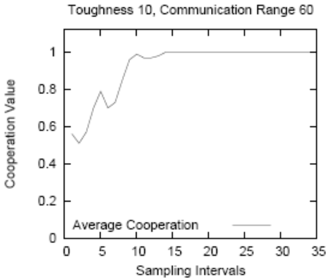
Figure: UAVs begin cooperative behavior to mass their fires on individual targets.

Experiments with the Parallel GMS Application

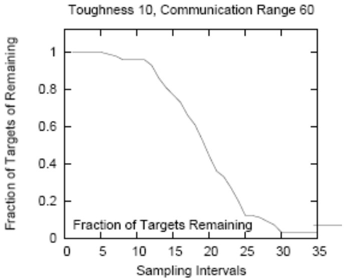
Objective: to provide an initial understanding of the potential of GMS as an S2 methodology.

- *Can the use of symbiotic simulation through ensembles of plausible models improve a physical system's performance?*

Face Validity



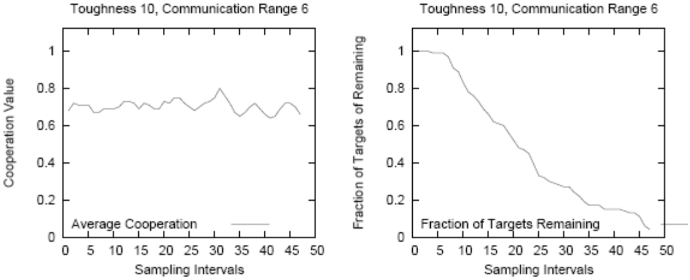
(a) Change in Average Cooperation



(b) Change in Target Population

Figure: Effect of Cooperation on Target Population with High Communication Range

Face Validity



(a) Change in Average Cooperation (b) Change in Target Population

Figure: Effect of Cooperation on Target Population with Low Communication Range

Experimentation with Emulator

- Used a system emulator, rather than an actual physical system
- Emulator objective is to minimize the number of time steps required to eliminate all targets.
  - the fitness of an individual is the average number of time steps required to eliminate all targets.
- Our position: Emulator's system would have an advantage over the system in a stand alone model. The controllable factors of Leader flags and Cooperation Thresholds can be modified by GMS to the benefit of the emulator

Experimentation with Emulator

Experiment	Average Time Steps	Standard Dev.
Emulator with GMS	1397.5	164.34
Stand-alone model	1777.24	129.68

Table: The results of the Parallel Application compared to the best performing Stand-alone Model. The Parallel Application provides significantly improved performance on average, but with higher variance.

Fitness of Emulator System Configuration

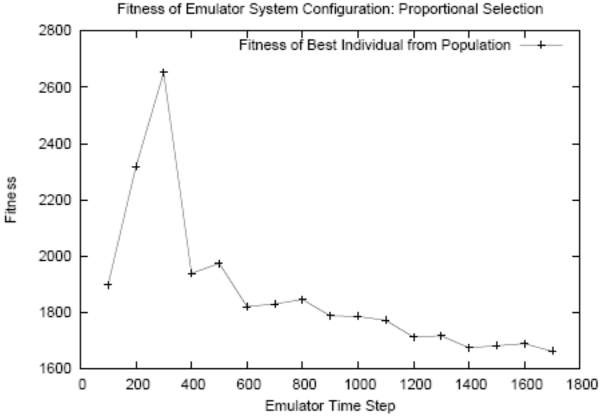


Figure: Fitness of Emulator System Configuration.

Concluding Remarks

- Valid, authoritative models of the physical system have limitations to cope with uncertainty (lack of information) in CPS.
  - GMS builds on earlier techniques such as Multisimulation and Exploratory Analysis, which experiment with an ensemble of plausible models.
- GMS involves a hybrid exploration strategy to study an ensemble of plausible models.

Concluding Remarks

- When parameterized to account for input uncertainty, GMS is able to dynamically update a system emulator resulting in improved performance.
- Initial study presents encouraging results, but has also left various problems unsolved.
  - Experimentation with additional GA designs can be performed to understand the features that make an Evolutionary Algorithm suitable for GMS.
- Other significant problems that remain: incorporation of Multiresolution Modeling, input analysis for estimating uncontrollable factor distributions, and handling of structural uncertainty.
- Given these challenges, autonomic GMS appears to be a rich opportunity for further study.